Incrementalizing Random Sketching for Solving Consistent Linear Systems

Vivak Patel
University of Wisconsin – Madison

October 2021

Coauthors & Reference

Vivak Patel, Mohammad Jahangoshahi, and Daniel A. Maldonado. "An implicit representation and iterative solution of randomly sketched linear systems." SIAM Journal on Matrix Analysis and Applications 42.2 (2021): 800-831.

A pre-print is also available on arXiv: 1904.11919.

Outline

1. Overview: Linear System & Randomized Solvers

2. Our Contribution

3. Our Procedure

4. Theory

5. Summary

OVERVIEW

Problem formulation

Our goal is to solve

$$\underbrace{A}_{n\times d}\underbrace{x}_{d} = \underbrace{b}_{n}, \tag{1}$$

for which we assume that at least one solution exists.

In particular, we are interested in randomized solvers, which recast the linear systems problem into a statistical estimation problem and then solve the estimation problem.

Two classes of randomized solvers

1. Random Sketching Methods. These methods use a matrix, M, with (much) fewer rows than A and solve the problem

$$\min_x \|(MA)x - (Mb)\|_2^2, \tag{2}$$

where M is a specifically structured random matrix, which we call a random sketching matrix.

Two classes of randomized solvers

2. Base Random Iteration. These methods use a random vectors, $\{w_k : k+1 \in \mathbb{N}\}$ and perform the iteration

$$x_{k+1} = x_k + \gamma A' w_k \left[(w_k' b - (w_k' A) x_k \right], \tag{3}$$

where $\gamma > 0$ is some scalar.

Why randomized solvers?

Random sketching methods have the lowest computational complexities for finding an ϵ-accurate solution (in residual) with high probability.

Why randomized solvers?

Random sketching methods have the lowest computational complexities for finding an ϵ-accurate solution (in residual) with high probability.

Base random iterations are cheap-per-iteration and converge linearly in the number of iterations for appropriately selected $\{w_k\}$.

Why randomized solvers?

Random sketching methods have the lowest computational complexities for finding an ϵ-accurate solution (in residual) with high probability.

Base random iterations are cheap-per-iteration and converge linearly in the number of iterations for appropriately selected $\{w_k\}$.

Even with high-probability, we would think that the best solution then is the random sketching approach as it is the fastest method available.

Challenges with Random Sketching

Challenges with Random Sketching

The number of rows of M depends on constants that are unknown and problem-specific, and is proportional to the reciprocal of ϵ.

Challenges with Random Sketching

The number of rows of M depends on constants that are unknown and problem-specific, and is proportional to the reciprocal of ϵ.

Therefore, MA, which can be cheap to compute (if M is sparse), might still be too expensive to construct and store!

TO SUMMARIZE

Random sketching has the best computational complexity, but we do not know how to choose the size of M and it is nontrivial to store MA.

OUR CONTRIBUTION

Overview & Consequences

We reformulate random sketching to implicitly construct MA and simultaneously solve the projected system (i.e., $MAx = Mb$).

Overview & Consequences

We reformulate random sketching to implicitly construct MA and simultaneously solve the projected system (i.e., $MAx = Mb$).

We do not need to decide on the size of M apriori. We can let the size of M grow implicitly, until, say, some stopping criteria is reached or the system is solved.

Overview & Consequences

We reformulate random sketching to implicitly construct MA and simultaneously solve the projected system (i.e., $MAx = Mb$).

We do not need to decide on the size of M apriori. We can let the size of M grow implicitly, until, say, some stopping criteria is reached or the system is solved.

Additionally, we do not need to create and store the matrix MA. We implicitly work with this matrix without constructing it.

BOTTOM LINE

Owing to **our reformulation**, we are able to move towards the **practical** use of random sketching methods to solve actual linear systems.

OUR PROCEDURE

Step 1: Streaming rows of M

Let $w_k \in \mathbb{R}^n$ denote the k^{th} row of a sketching matrix M. Our first requirement is to generate w_k on the fly.

Step 1: Streaming rows of $\boldsymbol{M}$

Let $\boldsymbol{w}_k \in \mathbb{R}^n$ denote the $\boldsymbol{k}^{\text{th}}$ row of a sketching matrix $\boldsymbol{M}$. Our first requirement is to generate $\boldsymbol{w}_k$ on the fly.

Example: Gaussian Sketch. $\boldsymbol{M}$ has independent, identically distributed standard Gaussian entries. Then $\boldsymbol{w}_k$ is simply an $\boldsymbol{n}$-dimensional standard Gaussian vector, and each $\{\boldsymbol{w}_j\}$ are independent.

Step 2: Iterative Solver

Recalling that $\{w_k\}$ are the rows of our sketching matrix M, we now work through the iteration

$$x_{k+1} = \begin{cases} x_k + S_k A' w_k \frac{w_k'(b - A x_k)}{w_k' A S_k A' w_k} & S_k A' w_k \neq 0 \\ x_k & \text{otherwise}, \end{cases} \tag{4}$$

and

$$S_{k+1} = \begin{cases} S_k - \frac{S_k A' w_k w_k' A S_k}{w_k' A S_k A' w_k} & S_k A' w_k \neq 0 \\ S_k & \text{otherwise}, \end{cases} \tag{5}$$

where $S_0 = I_d$ and x_0 is arbitrary.

Step 2: Iterative Solver

Note, $\{S_k\}$ are orthogonal projections onto the space perpendicular to the rows of MA that have already been observed.

Step 2: Iterative Solver

Note, $\{S_k\}$ are orthogonal projections onto the space perpendicular to the rows of MA that have already been observed.

In other words, $\mathcal{R}_l := \text{span}[A'w_0, \ldots, A'w_{l-1}]$ then S_l is an orthogonal projection onto $\mathcal{R}_l^\perp$.

Step 2: Iterative Solver

Note, $\{\boldsymbol{S}_k\}$ are orthogonal projections onto the space perpendicular to the rows of $\boldsymbol{MA}$ that have already been observed.

In other words, $\mathcal{R}_l := \textbf{span}[A'w_0, \ldots, A'w_{l-1}]$ then $\boldsymbol{S}_l$ is an orthogonal projection onto $\mathcal{R}_l^\perp$.

Therefore, as soon as we see the maximal possible linearly independent rows of $\boldsymbol{MA}$, then we will have solved the system to the highest accuracy possible allowed by $\boldsymbol{M}$ and $\boldsymbol{A}$.

THE CHALLENGE

How do we characterize this maximal set when the rows of *M* are generated on the fly and they can have an arbitrary (independent, random permutation, adaptive, dependent) structure to previously observed rows of *M*?

THEORY

Subspace Characterizations

Let $w \in \mathbb{R}^n$ be an arbitrary random variable. Define

$$\mathcal{N}(w) = \text{span}\left[z \in \mathbb{R}^d : \mathbb{P}\left[z'A'w = 0\right] = 1\right] \quad (6)$$

and

$$\mathcal{R}(w) = \mathcal{N}(w)^{\perp}. \quad (7)$$

Subspace Characterizations

Let $w \in \mathbb{R}^n$ be an arbitrary random variable. Define

$$\mathcal{N}(w) = \text{span}\left[z \in \mathbb{R}^d : \mathbb{P}\left[z'A'w = 0\right] = 1\right] \tag{6}$$

and

$$\mathcal{R}(w) = \mathcal{N}(w)^{\perp}. \tag{7}$$

Lemma

$\mathcal{R}(w)$ is the smallest subspace of $\mathbb{R}^d$ such that $\mathbb{P}\left[A'w \in \mathcal{R}(w)\right] = 1$.

Subspace Characterizations

Message: For an arbitrary random variable $\boldsymbol{w}$

- $\mathcal{R}(\boldsymbol{w})$ characterizes the row space of $\boldsymbol{w}'\boldsymbol{A}$

- $\mathcal{N}(\boldsymbol{w})$ characterizes the null space of $\boldsymbol{w}'\boldsymbol{A}$

-$\mathcal{V}(\boldsymbol{w})$ characterizes the deficiency of $\boldsymbol{w}'\boldsymbol{A}$ compared to $\boldsymbol{A}$.

Iterations to Maximal Set

Now for an arbitrary random variable w, let $\mathcal{R}(w)$ and $\mathcal{N}(w)$ be defined as before. For (a not necessarily related) set of random variables $\{w_k\}$, define

$$T = \min\{k \geq 0 : \operatorname{span}[A'w_0, \ldots, A'w_k] \supset \mathcal{R}(w)\}. \tag{8}$$

Iterations to Maximal Set

Now for an arbitrary random variable w, let $\mathcal{R}(w)$ and $\mathcal{N}(w)$ be defined as before. For (a not necessarily related) set of random variables $\{w_k\}$, define

$$T = \min\{k \geq 0 : \text{span}\,[A'w_0, \ldots, A'w_k] \supset \mathcal{R}(w)\}. \qquad (8)$$

Again, we have **not** imposed any relationship between $w, w_0, w_1, \ldots$. Therefore, T is quite generally defined. (This is useful when we consider parallel implementations.)

Convergence

Theorem

Let w be a random variable, and let $\{w_k\}$ be random variables such that $\mathbb{P}\left[A'w_l \in \mathcal{R}(w)\right] = 1$ for all $l \geq 0$. Define T as above. On the event $\{T < \infty\}$,

- For any $s \geq T+1$, $S_{t+1} = S_s$ and $x_{T+1} = x_s$.
- If $Ax = b$ admits a solution x^* (not necessarily unique), then

$$x_{T+1} = P_{\mathcal{N}(w)} x_0 + P_{\mathcal{R}(w)} x^*. \tag{9}$$

Do we solve the system?

Corollary

Under the settings of the preceding theorem, on the event $\{T < \infty\}$, $Ax_{T+1} = b$ if and only if $P_{\mathcal{V}(w)}x_0 = P_{\mathcal{V}(w)}x^*$.

Do we solve the system?

Corollary

Under the settings of the preceding theorem, on the event $\{T < \infty\}$, $Ax_{T+1} = b$ if and only if $P_{\mathcal{V}(w)}x_0 = P_{\mathcal{V}(w)}x^*$.

(1) When is $T < \infty$?

(2) When will $P_{\mathcal{V}(w)}(x^* - x_0) = 0$?

Do we solve the system?

Corollary

Under the settings of the preceding theorem, on the event $\{T < \infty\}$, $Ax_{T+1} = b$ if and only if $P_{\mathcal{V}(w)}x_0 = P_{\mathcal{V}(w)}x^*$.

(1) When is $T < \infty$?

(2) When will $P_{\mathcal{V}(w)}(x^* - x_0) = 0$?

Basically, when is this going to actually work?

When is this going to work?

Both of these questions will depend on how you choose $\boldsymbol{w}$, and how you design $\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots$ for your particular system. This should depend on the linear system's structure and the hardware environment.

When is this going to work?

Both of these questions will depend on how you choose $\boldsymbol{w}$, and how you design $\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots$ for your particular system. This should depend on the linear system's structure and the hardware environment.

We have simply stated a very general theory of convergence for such methods, and supply specific examples in the paper.

SUMMARY

We restated matrix sketching as a **random orthogonalization procedure** and characterized the convergence for **arbitrary** sampling methodologies. This allows us to **implicitly and incrementally** generate and grow *MA* without storing it explicitly.

THANK YOU

www.vivakpatel.org